Carnegie Mellon
Software Engineering Institute

Guidelines for Developing a Product Line Production Plan
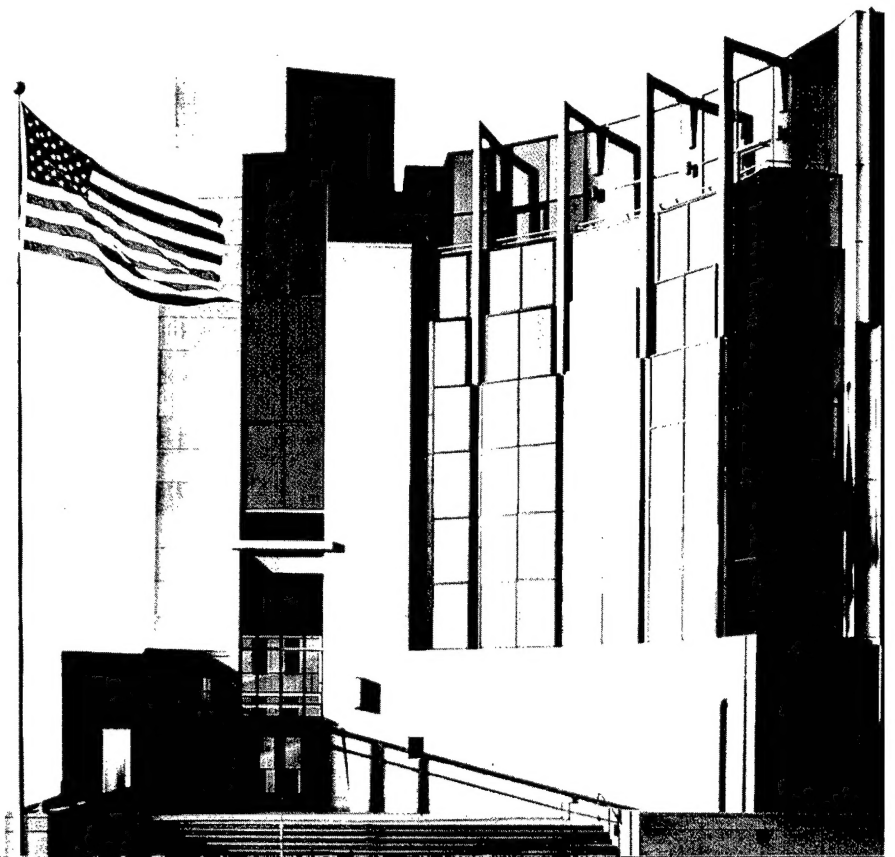
Gary Chastek
John D. McGregor

June 2002

20021009 037

TECHNICAL REPORT
CMU/SEI-2002-TR-006
ESC-TR-2002-006

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Guidelines for Developing a Product Line Production Plan

Gary Chastek
John D. McGregor

June 2002

Product Line Practice Initiative
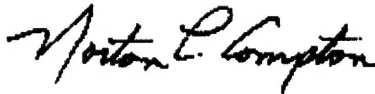
This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

[signature]

Norton L. Compton, Lt Col, USAF
SEI Joint Program Office

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Abstract

A production plan is a description of how core assets are to be used to develop a product in a product line. A product line organization creates such a plan to ensure that the correct core assets are used appropriately to build a specific product in a specific way. The production plans and techniques used to create products vary widely from organization to organization and from one product line to another. Because of this variance, the developers of production plans need some guidance about the plans' form and content.

This technical report provides guidance for creating, using, and evaluating a production plan. In addition, this report presents a classification scheme that describes the characteristics of a product line organization that influence the form and content of the production plan.

1 Introduction

The purpose of a software product line organization[1] is to create products. Organizations adopt a product line approach in order to achieve a number of goals [Clements 02a]. These goals include but are not limited to

- reduced time to market
- reduced production costs
- improved quality

A product line organization seeks to achieve these goals through an architecture-centric product development approach that achieves strategic reuse of assets. These assets include but are not limited to

- domain and requirements models
- the software architecture
- test plans and test cases
- reusable software components
- budgets, schedules, and work plans

The production plan for a product line captures the strategy for developing products from the core assets. The production strategy is a key driver of the design of the core assets. The core-asset developers create the strategy while the core assets are being created. By defining the product development process, the production strategy specifies the "prescribed manner" of development called for in the definition of a software product line [Clements 02a]. The core asset developers are responsible for creating the production plan that will communicate the production strategy to the product developers.

The product developer is the person (or people) responsible for the creation of a specific product in the product line. The product developers create a product-specific production plan from the general production plan created by the core-asset developers. The product develop-

---

[1]   A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements 02a].

ers may also be responsible for specifying the product requirements, customizing the product line architecture and components, and tailoring the testing assets for the specific product.

To maximize the benefits of the product line approach to an organization, several issues related to product creation must be considered during the creation of the production plan:

- What is the most efficient organization of the core assets for product building?
- How can core-asset creation be coordinated to support consistent and effective product development in a product line?
- What information about the core assets would be most helpful to the product developers?
- What variation mechanisms do the core assets provide?
- How can the product developers efficiently utilize the variability mechanisms in the core assets?
- How much flexibility should the product developers have in modifying the core assets of the product line?
- Where can help be found when specific problems arise during integration of assets?
- How can the specific product requirements be used to estimate cost and schedule?

This remainder of this section provides an overview of the concepts that are covered in this report. Section 2 describes a useful classification scheme for product lines that explains some of the variation from one production plan to another. Section 3 describes the general approach to creating the production plan, and Section 5 presents techniques for tailoring the production plan for a specific product. Section 4 describes the product development process. Section 6 provides guidance on using the production plan, and Section 7 provides information on evaluating the production plan. Section 8 describes future work to be done on the topic of production plans.

Core-asset developers should read Sections 3 and 4 to understand how to create the production plan and the product development process. They should also read Section 7 for information on evaluating the production plan. Product developers should read Section 5 to understand how to tailor the production plan to create a product-specific production plan. They should also read Section 6 to understand how to use the plan, and Section 7 for information on evaluating the product-specific production plan.

1.1 Production Plan

The products in a product line are built from the product line's core assets which include the requirements, architecture, components, test cases and plans, schedules, and budgets. Each core asset has an attached process that is created by the core-asset developer and that de-

scribes how the core asset is used in product production. The production plan is a description of how the attached processes cooperate to yield a product [Clements 02a].

The production plan captures how the product line organization builds any product. The plan coordinates the efforts of managers, product developers, testers, and clients. The plan links together the information provided by the product requirements, business case, architecture description, component specifications, asset-use processes, and other sources, such as user manuals.

The production plan expands on the Technical Considerations chapter of the Concept of Operations (CONOPS) [2] by providing a more complete description of the process by which products are created [Cohen 99]. The production plan specifies the following:

- inputs needed to build a product
- activities that result in a completed product
- roles and responsibilities of the product developers
- interactions needed with other groups in the organization
- schedule and resources associated with building the product

The production plan for a specific product is created from the production plan for the product line. In some product lines, one plan fits all products. With an automatic generation strategy, each product is built automatically from a completed checklist of product features, and the product-build process is trivial; the production schedule is almost instantaneous, and only the production resources are a real issue. In this case, the production plan contains all of the information needed by the product developers.

In other product lines, the product developers instantiate a product-specific production plan from the production plan [Clements 02a]. The product-specific production plan is based on the choices made at the variation points in the architecture. Each choice imposes constraints that span such concerns as delivery dates for components, licensing fees, cost estimates for the product, and availability of personnel with specific expertise. As such, each product-specific production is unique.

The product-specific production plan contains only the information that is relevant to the creation of that product. As choices are made at specific variation points, the production plan is tailored to include only the processes and resources required by the selected variations. For

---

[2]   The CONOPS document is often used to describe how a computer system will be managed and operated. In product line organizations, it is used to describe how the product line organization operates. The CONOPS helps personnel understand the roles and responsibilities in the organization.

example, a product line may offer basic and deluxe products, where the deluxe products have special security features. The product line architecture would have a corresponding variation point that permits different operating systems: the basic products are built using a non-secure operating system, whereas the deluxe products are built using a secure operating system. Once the product to be built is identified as deluxe, assets that are related only to the non-secure operating system are irrelevant. The product-specific production plan that results from the tailoring is a concise guide to building the one specific product.

The production plan is a core asset of the product line. Like any other core asset, the production plan has its own attached process that includes activities from planning through product development. The attached process of the production plan defines the glue that binds together the other attached processes. It includes procedures for estimating the size of the product that will result from combining the selected core assets. The attached process uses the size estimate to determine the time and resources required to create the product. This information is input into an activity that produces the schedule for the activities described in the production plan. The exact algorithms for size estimation and activity scheduling vary from one product line to another and are beyond the scope of this report. This report focuses on the creation of the product line's production plan, product-specific production plans, and the interactions of these plans with other core assets.

1.2 Creating the Product Line Production Plan

The production plan for a product line covers a wider range of topics and is more complex than the typical project plan used by single-product projects. While the exact form and content of the production plan varies from one product line organization to another, the plan is nonetheless a means of communication between the core-asset developers and the product developers, as well as a source for resource and schedule estimates.

Production plans in hard-goods manufacturing include the sequence of activities needed to build a product, schedules of activities, bills of materials, and assignments of roles and responsibilities [Hax 87]. However, a production plan in hard-goods manufacturing must account for the actual building of multiple copies of the physical product. The major effort in building a software product is expended only once; additional copies are automatically reproduced. The software production plan often does not consider the creation of physical copies of the product in the schedule or cost estimates.

Although the core-asset developers have primary responsibility for developing the production plan, the product developers contribute as well (see Figure 1):

- The core-asset builders contribute to the production plan from the perspectives of having analyzed all products within the scope of the product line and having developed the core assets. They are responsible for including sufficient information about each core asset to

allow the product developers to understand the assets and make informed choices. The core-asset builders provide guidance to the product developers on how the assets should be used by attaching a process to each core asset. For example, the attached process of the software architecture provides a technique for tailoring the architecture to fit the specific product.

- The product developers contribute to the production plan from their perspective of actually executing the product-building process. Product developers provide feedback to the core-asset developers initially as they attempt to understand the product-building process. The product developers later provide feedback based on their experience with the product-building process. The product developers identify process defects, unrealistic constraints, and implicit assumptions in the processes attached to the core assets. The product developers also identify interactions between independent processes that are not properly coordinated and contribute to evolving resources such as FAQs, lists of heuristics, and patterns catalogs that are derived from actual experience.
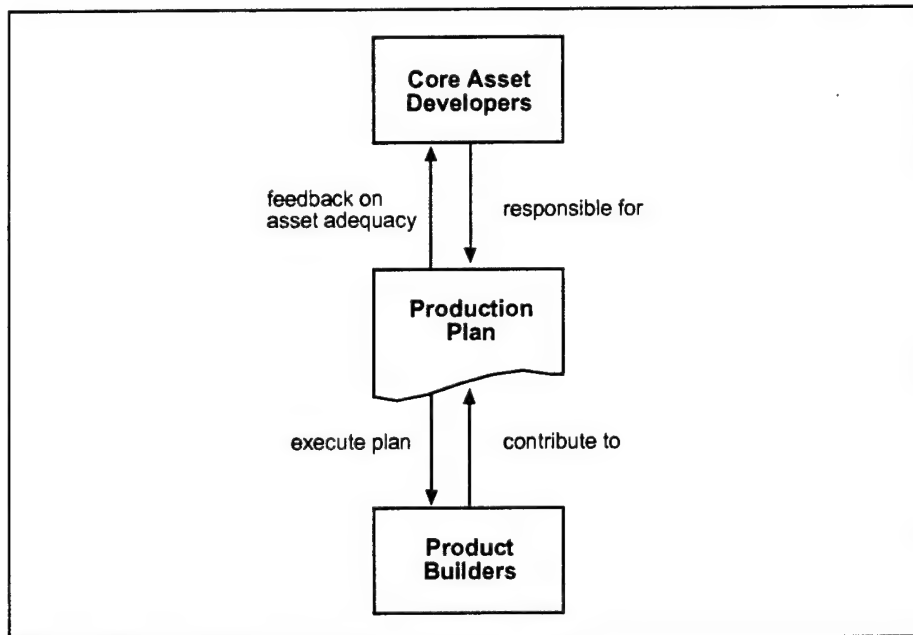
Figure 1:   Relationships Between Core-Asset Developers and Product Developers

The production plan is an implementation of a production strategy. This strategy is a key driver in the design of the core assets; it determines exactly how the core assets are selected for use in building a specific product and how the attached processes of the selected assets are coordinated. The exact strategy used by a product line depends on a number of factors including organizational factors that will be examined in Section 2 and technical factors that will be examined in Section 3. The strategy must provide ways to sequence the activities defined in the core assets' attached processes and to resolve conflicts between them.

The production plan can take many forms. If the product line has a fixed set of possible requirements, the production process can use automated checklists of requirements to configure

build scripts. For more manual approaches, the production plan provides mappings that coordinate the attached processes of core assets. Mappings may go from the requirements to other assets, and from the variation points of the product line to sets of requirements. Other mappings may include a correspondence between architectural patterns and specific groupings of requirements, or between clusters of components and architectural patterns.

Sections 3, 4, and 5 will explore these ideas in greater detail.

1.3 Using the Production Plan

The product developers use the production plan as a guide from product inception to product delivery along the most efficient path possible. The plan provides information that allows technical management to track the progress of product development.

The information in the production plan is presented in two distinctly different views:

- The product developers need a general understanding of the core assets and how they are used to construct products. This overall perspective is an important first step, which can be provided in an overview section in the plan. As product development proceeds, the developer follows the development process in the production plan. At times when the plan allows the developer to make choices between variants in a core asset, the overview knowledge helps the developer determine the best ones.

- The product developers also need specific, detailed information about the core assets that pertain to the product under construction. For example, when using a component, the parameters need to be set for a particular product. The production plan does not contain detailed information about each asset; rather, it contains pointers to it. The product developer needs fast and efficient access to this information. The production plan provides mappings between sets of assets. For example, given a specific set of requirements, the product developer can use the plan to determine which assets are needed to develop the product defined by those features.

The process attached to the production plan guides the product developer through the development steps beginning with product planning and ending with product release. Early activities identify the variations that uniquely define the product, select the core assets, and create initial information such as the bill of materials. Later activities use the attached processes of the selected core assets to drive product development. Section 6 will explore these ideas in greater detail.

2 Relevant Characteristics of Product Lines

Product lines differ from each other in many ways [Clements 02a], particularly in those that affect the production of products in the product line:

- practice area expertise
- maturity of the product market
- automation of product creation

The three orthogonal axes shown in Figure 2 represent these factors. Each axis can be viewed as a continuum. The labels on each end of the axes provide examples of "extreme" values. The factors are useful for analyzing differences between production plans, but they are not sufficient to constitute a formal (complete) taxonomy of product lines or production plans.

There are other factors that vary among product lines, for example, the culture of the organization. This culture usually determines how formally the production plan is written. How the product line will operate is a second factor. The CONOPS may specify a mode of operation that reduces the options for how the product line operates, but it does not specify the exact strategy to use [Clements 02a]. These factors may be included in the classification scheme later, if further investigation shows a relationship.

2.1 Practice Area Expertise

The degree to which the organization has institutionalized a product line approach, as described in the practice areas in *Software Product Lines: Practices and Patterns*, is an important influence on the production plan [Clements 02a]. The organization that has developed deep expertise in a broad set of product line practices is at one extreme. On the other extreme, the organization is not ready to adopt product line practices. A product line organization between the extremes is systematically improving its expertise in selected practices and still evolving the core assets related to those practices.

A product line organization that has institutionalized the product line practices will have a production plan that prescribes the production process in detail. The organization that is not ready to adopt product line practices has an informal production plan that is often distributed

among several documents and may simply be in the minds of the developers. As practices are institutionalized, the expertise is captured in assets, and plans become more complete.

2.2 Market Maturity

A mature market is characterized by relatively stable feature sets for the products in the product line. There is agreement on terminology, and standards have been established in the relevant domains. There is little differentiation between products from competing organizations. At the other extreme, an emerging market is characterized by products that have rapidly changing—and usually expanding—feature sets. In this case, there are inconsistencies between the terminologies used by various customers. For a market between the extremes, the feature sets are expanding. Some in-place standards are occasionally replaced, causing radical shifts in the component inventory and in the architecture.

The production plan for products in a more mature market can be more detailed than one for a rapidly changing market. There will be meta-information, such as design patterns, that have emerged out of multiple product development efforts. Immature standards and technologies go through many versions early in the market cycle. Initial products in the product line incur much rework until abstractions can be clearly and precisely defined. For example, product lines in insurance software are more stable than product lines of wireless devices.

Highly correlated to this dimension, but not sufficient to be a separate dimension, is the maturity of the content domains used in the product line. A market cannot be mature unless the underlying bodies of knowledge—the domains—are stable and well defined. A domain that is rapidly changing due to research progress corresponds to markets that are also rapidly changing. In mature domains, the production plan can point to external sources of information that explain standards and concepts.

2.3 Automated Product Creation

A product line in which product creation is highly automated constrains the product developer by offering a fixed set of choices from the set of available features. In this context, the product developer does not need to know much about the domain or about the actual components being used to implement the product. For a product line organization at the other extreme, each new product requires that a new system build script be created. In this case, the product developers need extensive knowledge of the components as well as the domain.

The range of product developer roles corresponds to the range of possible production strategies. That is, in a product line where product creation is highly automated, the product developer could be a single developer who simply identifies the product to be developed by select-

ing the features that correspond to the appropriate variation choices. For product lines that use completely manual creation, the product developers could include requirements analysts, architects, component builders, testers, and so forth.

A product line organization in between these extremes has some automated support for creating a new product from the core assets, but the process is not fully automated. For example, Cummins Inc. uses 20 builds for over 1,000 products [Clements 02a].

The production plan for the automated product line describes each parameter to the build process and the possible values for each of those parameters. In the product line where product creation is manual, the production plan provides detailed information about the available components and provides instructions for creating new build scripts.
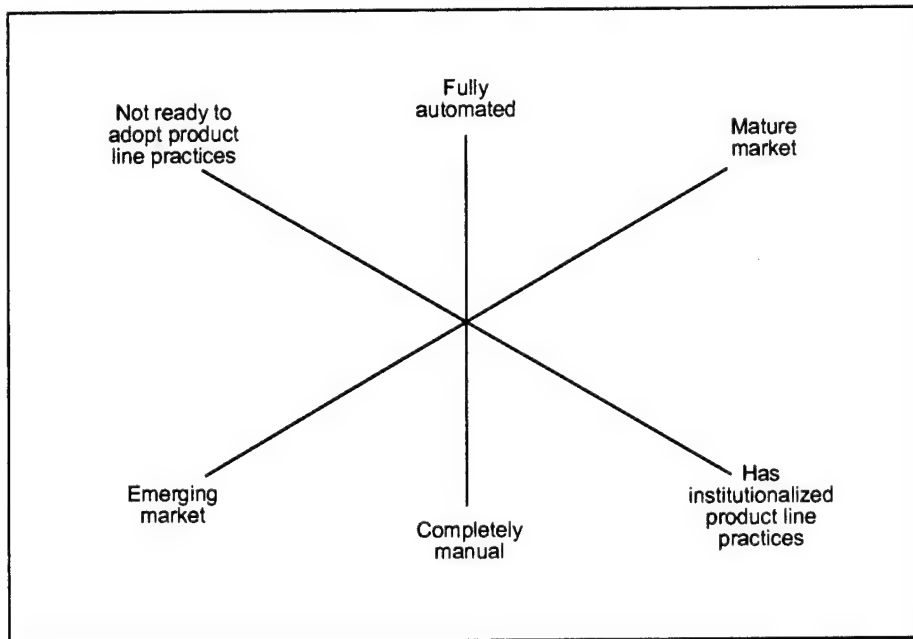
Figure 2: Classification Dimensions

2.4 Classification of a Product Line

The three dimensions discussed above provide a rough means by which we can characterize production plans. The intent is to be able to talk about the variations from one production plan to another by giving a relative position along a continuum. Such a classification enables readers to locate their organization along the continuum and relate the discussions in this report to their product line organization. Figure 3 illustrates the directions along the continua in which product lines are likely to move over time. When the product line moves, the production plan must be changed to accommodate the line's new position. The rest of this report will

discuss the modifications that should be made to production plans to accommodate these changes.
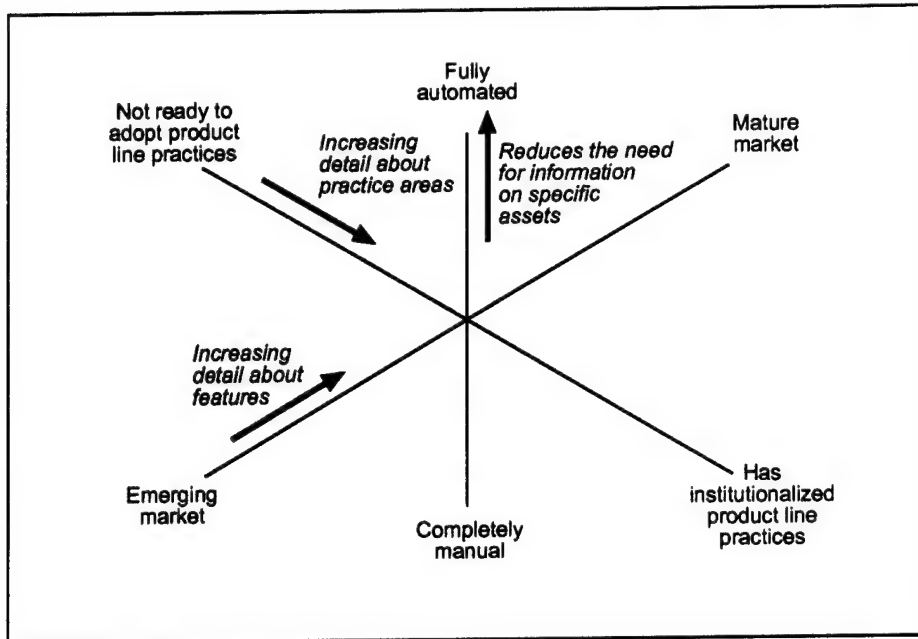
Figure 3: *Product Line Evolution Along the Classification Dimensions*

3 Issues in Building a Production Plan

The purpose of this section is to explore the issues associated with building a production plan. As described in Section 1, the product line's production plan documents the product development process, which can be fully automatic, semi-automatic, or completely manual. In any case, the product line has a strategy for creating products, and that strategy determines the development process that is documented in the production plan.
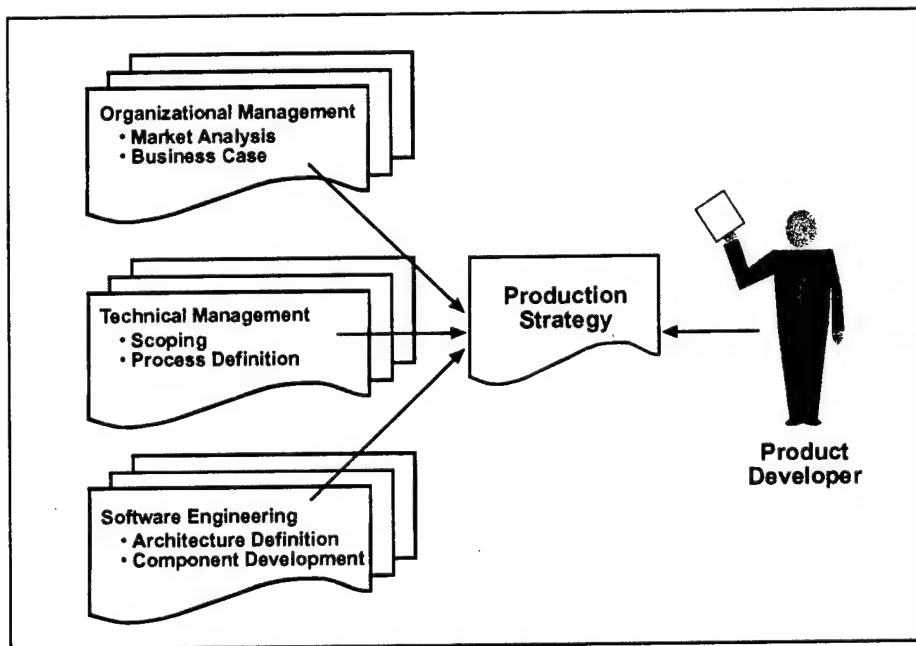
Figure 4:   Production Strategy

Figure 4 illustrates that there are a number of inputs to the production strategy, including market analysis, scoping, technical issues, and the business case. As users of the production plan, the product developers are a source of requirements for the production strategy. This strategy is, in turn, the primary input to building the production plan.

Viewing product production as a strategy leads to a number of questions:

- What goals must the strategy achieve?
- What qualities must the production strategy possess?
- When should the production strategy and plan be developed?

- What is the environment in which product developers work, and what problems do they face that can be addressed by the production plan?
- How are core assets and their attached processes coordinated by the production strategy?
- Which core assets are addressed in a production plan?

This section will address those questions. Section 3.1 describes the production strategy and its early developmental influences on the production plan. Section 3.2 describes the product developer's view of a product line. Section 3.3 discusses creating the production plan.

3.1 Production Strategy

The production strategy coordinates the design and use of the core assets. It begins as an informal notion, evolves concurrently with the core assets, and is ultimately documented in the production plan. The production strategy is based on the product line goals and influenced by the technologies to be used during production. This strategy specifies techniques and conditions for product development that support those goals.

The production strategy defines a number of aspects of development, including

- the expertise of the product developers
- how the product developer identifies the product to be built
- the product development process
- the technical environment used to build the software products

3.1.1 Qualities of the Production Strategy

The production strategy should possess qualities[3] that ensure that the production plan supports the goals of the product line. The qualities for the production strategy, which come from the business case, are identified before the strategy is defined so that they can be incorporated into the strategy as it is developed, rather than added as an afterthought. The following list provides examples of qualities, the ways in which they might be realized in the strategy, and their ultimate effect on the production plan.

- flexibility - The product line has the goal of adopting emerging technologies as quickly as possible. This could be realized in the production plan as a set of mappings that trace relationships between overarching technologies such as the process distribution model and those specific components that are dependent on that technology.
- simplicity - The product line has a goal of reducing personnel costs through the use of non-technical, or less technical, product developers. The strategy is to hide as many de-

---

[3] There is a distinction between the qualities of the products themselves and the qualities needed for the production of those products.

tails about product development as possible from product developers. The production plan presents a high-level production process with few choices and provides only the required information about the individual core assets.

- performance - The product line may have a goal of increasing the speed at which the company enters new markets. The production strategy would be to produce usable products with minimal functionality as quickly as possible. The production plan would define an incremental plan where a product would be produced over multiple releases, each with a larger portion of the product's envisioned functionality.

- modularity - A product line in an emerging market has the goal of maintaining currency with evolving standards. The strategy is defined in terms of the individual core assets. While it ensures consistency among the assets, it does not modify the attached processes of the core assets in any way. Assets may be replaced or modified without affecting other assets. The production plan contains pointers to core assets and their attached processes; it does not integrate them into a single asset.

3.1.2 Influences on the Production Strategy

A number of factors influence the production strategy. The position of the product line along the dimensions described in Section 2 determines part of the production strategy. The organizational management practice areas of "Market Analysis" and "Business Case Development" have the earliest and most significant influence on the production strategy. However, the technical management practice area of "Scoping" and the software engineering practice areas of "Architecture Definition" and "Component Development" also influence the production strategy. [4]

The market analysis can drive the production strategy. The following considerations expand on the market dimensions introduced in Section 2:

- The market may be emerging and in flux, the products in that particular product line may be volatile, and the product features may be rapidly changing. A greater understanding of the domain may be necessary to support automatic generation of products. Such complex products in an immature domain may require substantial customization at product-build time.

- The market may be mature, and the products and their features may be stable. In such a mature and well-understood domain, the automatic generation of products can be viable.

- The market might be highly competitive, and a rapid time to market may be required for an organization to compete successfully. This situation could favor a more automatic generation of the products in the product line.

- The market might be competitive, with multiple, highly demanding customers, each with special needs. If the domain is relatively stable, the basic products can be generated automatically and then customized to the particular customer's specifications.

---

[4] This section describes how the practice areas influence the production strategy. The production strategy will also influence how those practice areas are realized.

- The business case can drive the production strategy. For example,

  - An organization may wish to reduce its long-term cost of software development by reducing the number of software developers required to produce a particular set of products. This could lead that organization to adopt the automatic generation of products if the domain is suitably mature and the organization has institutionalized the required practice areas.

  - An organization may have an understanding of both the emerging software market and the expertise required to develop the products in a product line. Keeping that market and software expertise within the organization may be a key market advantage, and the production strategy may support that goal by providing opportunities that challenge the developers of new products.

  - An organization may be adopting a product line approach for the first time. Current staff may be apprehensive of, and resistant to, the change. The successful adoption of a product line approach may be a longer-term organizational benefit, so a less-dramatic change to the product-building process for the first project may be appropriate. The production strategy may include a modified version of the existing development process.

Other questions addressed by the business case that may potentially affect the production strategy include assumptions about the types of development resources that will be used in the product line and policies about the acquisition of commercial assets.

Other practice areas can influence the production strategy:

- The "Product Line Scoping" practice area (i.e., determining the types of products to be built) can drive the production strategy. For example, if the scope of the product line includes products that have very tight performance requirements, such as hard real-time systems with response times that push the limits of the hardware, the production strategy must provide for individual product customization at product-build time. If the domain is relatively stable, then the basic products could be generated automatically and then customized to the particular customer's performance requirements.

- The "Process Definition" practice area influences the process model chosen as the basis for the production process of the strategy. Whether the process model is waterfall (only for small, largely automatic processes), iterative-incremental, or agile, the assumptions and requirements of the process model enhance certain properties of the strategy and degrade others. An agile process model may enhance the performance of the production process.

- The "Architecture Definition" practice area affects the production strategy. One of the architecture's quality attributes is buildability [Bass 98]. The mechanisms chosen to achieve this quality will affect the strategy for building products. If the product line has a goal of using existing technologies and assets, the strategy will define a production process that uses available assets.

- The "Component Development" practice area influences the production strategy. In particular, the information provided in the specification of each component determines how much reasoning the production process is able to do when selecting assets. A specification that does not include information about performance and other qualities of the component may prevent the use of certain tools and the automatic configuration of products.

3.1.3 Interactions Between the Production Strategy and Core Assets

The production strategy is a key driver of the form of the core assets. This implies that development of the strategy begins early in the life of the product line. The strategy provides direction to the core-asset developers to ensure that their individual pieces contribute the appropriate information to the production process. For software components, for example, the strategy defines the structure of information that each component should make available to the product developers. This might include a standard set of interface definitions and a tool set for examining and comparing components for compatibility.

The form of the strategy can be affected by the choice of core assets. The need to align with corporate mandates such as a common tool suite, language, or style can also have an impact on the strategy. Creating a production strategy is a process of balancing business goals against the reality of existing software development practices. This implies that the development of the strategy continues as long as new core assets are being selected.

The strategy defines how the product developers interact with the core assets. Product developers may interact with the core assets through a specially constructed development environment, a commercial product line tool, or word processors and individual programming tools. If the strategy is to use non-technical product developers, specialized, robust environments will be necessary. Even highly technical product developers can benefit from tools that associate core assets automatically.

3.2 Product Developer's Perspective

The product developers are the users of the production plan and all of its parts, including the product line's core assets, attached processes, production process, and production strategy. Hence, these artifacts should be designed to satisfy the needs of the product developer.

We can further define what a production plan should be and how it should be structured by examining the tasks required of the product developers and by considering the environment in which they work. The product developer needs to

- identify the product to be built
- identify the specific assets required to build that product
- perform any necessary customization of each core asset not addressed by its attached processes
- integrate components

The product developer needs the production plan to be

- efficient. All activities in the production process are required to produce the specific product being developed.
- complete. All information that is needed is in the production plan.
- understandable. The information in the production plan is usable without outside assistance.
- usable. The product developer is able to locate needed information quickly and easily.

The core assets can be separated according to whether they are used directly by the product developer. If the products are generated automatically from the product features, the requirements model may be the only asset to which the product developer needs access. If the products are hand-customized from the core assets, the product line requirements model, architecture, and components must all be available to the product developers. The business case and the market analysis, for example, are core assets that are seldom needed by the product developers.

3.3 Building the Production Plan

The previous sections have described the goals, qualities, and audience for a production strategy and production plan. This section will describe how these factors come together to shape the production plan.

3.3.1 Plan Structure

A basic outline of the production plan is

1. Introduction
   Production context
   Audience
   Qualifications

2. Strategic view of product development
   Assumptions
   Qualities
   Products possible from available assets
   Production strategy

3. Overview of available core assets
   Basic inputs and dependencies
   Variations

4. Detailed production process

5. Tailoring production plan to product-specific production plan
   Product production

6. Management information
   Schedule
   Production Resources
   Bill of materials
   Product-specific details
   Metrics

This outline illustrates only the basic contents of the production plan. The order of items in the outline is not particularly important; because of mutual dependencies, the plan is created iteratively and incrementally. The core-asset developers provide sections such as the strategic overview. The product developers add sections, such as the product-specific details, as part of the tailoring process. The product identification step in the production process adds to the basic bill of materials.

The production plan evolves the production strategy and expands the strategy's notions of schedules and resources into more complete definitions. The plan combines the production strategy and the required core assets into a production process with a set of activities, schedule for the activities, and required resources (people and bill of materials). The activities implement the production strategy. The resources required are determined by the basic strategy and the variation choices made in defining the product. The schedule combines the activities and the resources in order to sequence the activities and allocate resources to execute the activities. The production process is discussed in more detail in Section 4.

The production process is the production plan's attached process and includes activities in which product-specific details are added to the product line production plan to create the product-specific production plan. The activity that guides the tailoring of the production plan is product identification, discussed later in this section; the pieces that are created as a result of the tailoring of the production plan are discussed in detail in Section 5.

3.3.2 Core Assets

The production plan presents the core assets to the product developers at the appropriate places in the production process. This presentation involves highlighting the core-asset de-

---

tails that are relevant to the product developer and organizing the core assets based on the product-variation points. These points are identified by the core-asset developer as the places in that asset where the products differ. For example, in a feature model for a product line, an optional feature is a product-variation point [Kang 90]. Some of the products will have that feature, but others will not.

Taken as a group, the core assets are designed to be comprehensive to ensure that all the products in the product line are addressed by those core assets. As the individual core assets are being built, the decisions relevant to product variations are scattered throughout those core assets. In other words, the core assets are organized as a tuple (core asset, product-variation point, distinguishing product characteristics,[5] and instructions[6]). Each core asset's product variation points are identified, and for each one, ways of tailoring the core asset for the specified product are described.

The production plan is organized to reduce the effect of that scattering on the product developer. This is accomplished by

- considering only the assets needed by the product developers (see Section 3.2) in the production plan
- organizing and presenting the assets using a tuple such as "product identifier, core asset, product variation point, instructions" in the production plan

For example, under certain conditions, product identification can be based on the product line features. The features that distinguish the products can be organized to minimize the number of questions needed to determine which product is to be built. An organization might choose to package certain features together. In this way, if a customer wants a particular feature, that customer must select from the packages that contain that feature. In that case, product identification can be based on selecting a set of packages.

If there are many distinct products, product identification becomes more complex. The production plan must identify products based on additional characteristics, including quality features. The tuple may be expanded to include the extra information.

Details of the architecture that are not important to the production process can be hidden from the product developer. For example, if security is not an issue for a specific product, that view of the architecture can be hidden from the product developer.

---

[5]   The distinguishing product characteristics provide a way of selecting from the alternatives of the product-variation point. They can be product identifiers or characteristics of a class of products.
[6]   The instructions describe how to customize the core asset for the specified product at that particular product-variation point.

4 Describing the Product Development Process

The production plan describes the process for building a product in the product line. The entry criteria for this process are defined in the CONOPS and typically require that the product planning and approval activities be completed prior to building a product. The product-building process is described using the process definition style used for other processes in the organization.

The Product Builder Pattern described by Clements specifies a set of product line practice areas that are used in building a product in a product line organization [Clements 02a]. Each practice area is a body of work or a collection of activities that an organization must master to carry out the essential work of a product line. Figure 5 shows the practice areas[7] used in the pattern and the interactions between them.

For a specific product line, the practice areas necessary to build a product depend on how well the organization has institutionalized the product line practices, the maturity of the market, and the degree of product development automation (see Figure 2). These practices comprise many activities that can be blended to form many different processes. For this reason, no single, specific, product-building process is defined in this report. The discussion and examples will remain at the level of practice areas.

This section describes three examples (see Table 1) that correspond to product lines at specific points in the classification shown in Figure 2. The descriptions of these examples are based on the Product Builder Pattern and its variants.

Table 1:   Examples

Example	Market	Practices	Process
1	Immature	Not Institutionalized	Manual
2	Mature	Institutionalized	Automated
3	Mature	Institutionalized	Semi-automated

---

[7]   The appendix describes the complete set of practices used in the Product Builder Pattern. The SEI's *Framework for Software Product Line Practice* provides more complete descriptions.

4.1 Example 1

In this section, an example process is presented for an organization that has not institutional-ized the practice areas, that builds products in an immature market, and that has only mini-mally automated its product-creation process.

As described in the Product Builder Pattern, this product line requires expertise in all of the practice areas shown in Figure 5 [Clements 02a]. Depending upon the launching strategy, either the first few products will be built from immature assets, or the product teams may ac-tually create some of the assets as they build the product. The functionality of products changes as competitors rapidly add new features to gain market share. Implementations of components can be replaced quickly to improve quality.
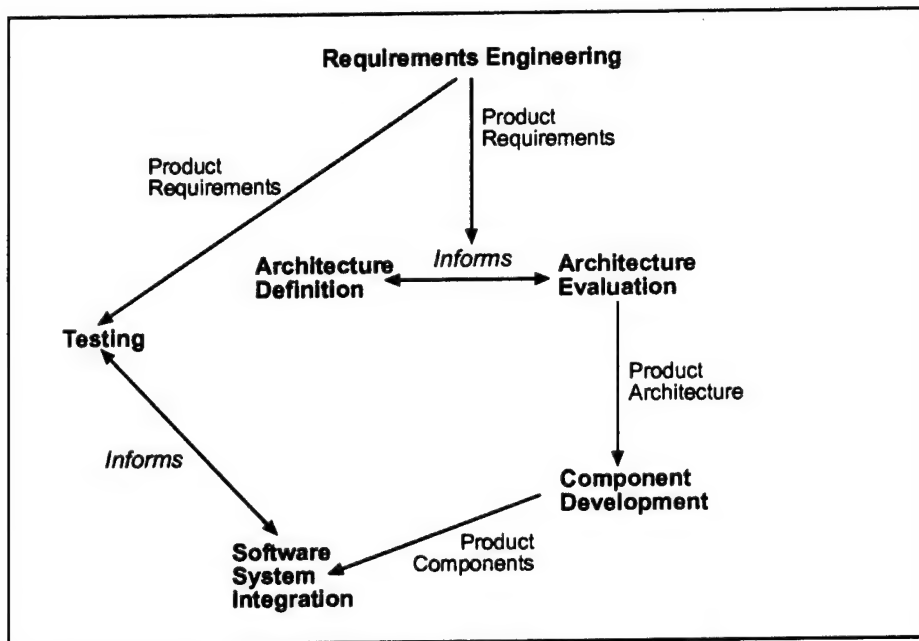
Figure 5: Dynamic Structure of the Product Builder Pattern[8]

The production plan plays a critical role in this type of product line. The plan must be com-prehensive because product developers are, at first, unaccustomed to their roles. It must be modifiable and extendable, because there will be many changes and additions. There is a dy-namic tension between the need to be sufficiently detailed to guide the product developer through rapidly changing procedures and the amount of resources it takes to keep the plan current with the latest procedures. This tension may be resolved somewhat by using dynamic, Web-based documents that can be updated easily and are available for reference rapidly.

---

[8]   The Each Asset Pattern describes how the practice areas are applied to develop the core assets [Clements 02a].

The product-build process is highly iterative in this type of product line. New releases of assets occur more frequently, and the differences between versions are more numerous than those for more mature product lines. The production plan establishes criteria for each step in the process by which decisions are made to either iterate back to previous activities for more refinement or to proceed forward to the next activity. The arrows in Figure 5 describe the allowable paths between practice areas.

4.2 Example 2

In this section, an example process is presented for an organization that has institutionalized the product line practice areas, whose product line resides in a mature market, and where the product development process has been automated.

As described by the Product Gen variant of the Product Builder Pattern, the product development process is highly automated, hence no new development is required [Clements 02a]. The product developers collect requirements, use a product line tool to indicate the requirements to be used, provide required parameters to the product-build tool, and test the resulting product. This simplified process flow is shown in Figure 6. Product development techniques of this type are discussed by Batory and Weiss [Batory 97, Weiss 99].
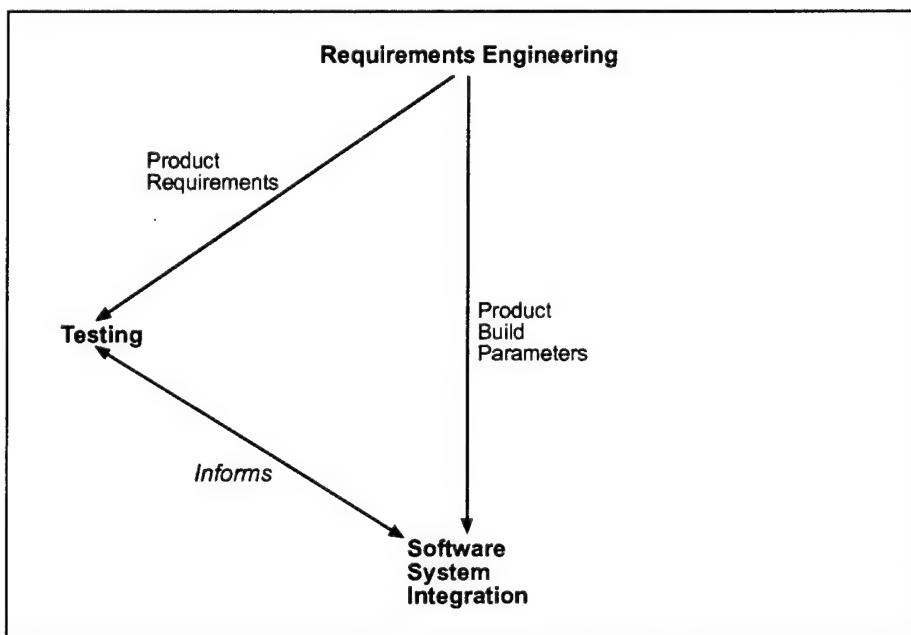
Figure 6: Dynamic Structure of the Product Generation Variant

In this type of product line, the requirements elicitation process consists of selecting from a fixed set of features. The requirements analysis process is automated and performs consis-

tency and completeness checks on the selected set of requirements. After the requirements are selected, the build tool automatically constructs the application.

Once the requirements checklist is completed, the set of system test cases has also been determined. The system test cases are associated with specific requirements and are added to the test suite as requirements are selected. The test cases focus on interactions among the assembled components. Automated testing tools vary the values of parameters within specific bounds to maximize the coverage of the product. The test reports are retained as assets in order for the ongoing computation of reliability to be recalculated as the amount of test coverage increases.

The production plan for this product line is basically the documentation for the requirements set, including definitions and dependencies. The plan also includes the instructions for using the requirements and testing tools to develop a product. The bill of materials includes any external components that incur royalty fees so that the product-specific production plan provides a unit cost for the product.

4.3 Example 3

In this section, the development process for a mature product line organization that is building products for an evolving market is presented. The product-creation process is automated in the areas of requirements engineering and system integration. Activities such as architecture definition and testing still depend upon the expertise of the personnel.

As Figure 5 shows, all practice areas may be needed to produce a specific product in this product line and are included in the production plan. The product-specific production plan includes only those practices that are required for that product.

The production plan for this type of product line is a rapidly changing document, but the change is well managed. Most changes to the production plan are not propagated to the product-specific production plans where production is in progress. There is no time to change the way the product is being created unless the changes correct fatal flaws in the production plan.

5 Specializing the Production Plan for a Specific Product

The production plan takes on many different forms as illustrated in Section 3. The plan may simply guide the product developer through a predetermined, unchanging, product-build process that fits all the products that can be produced in the product line. More likely, the product-build process varies depending upon which features are selected. In these cases, the production plan is designed to be specialized to become a product-specific production plan for each product that is built. The discussion that follows assumes the need for this specialization.

The core-asset developers create a production plan as one of the core assets for the product line as described in Section 3. The product development team then specializes the production plan for that specific project.[9] The process attached to the production plan guides the product development team in creating the product-specific production plan.

That attached process provides guidance on how to

- select and order the process steps that are needed, based on the product definition
- develop the product's bill of materials listing all of the assets that will be used for this specific product
- create the cost estimates and time schedules for building the product
- tailor the parts of the product plan's core asset that must be changed

The output from this activity is the product-specific production plan.

5.1 Selecting and Ordering Process Steps

The variations between products are realized in different requirements, different dependencies between portions of the system, asset tailoring, additional assets, and possibly different implementation technologies. As the product team makes decisions about specific variations, the product takes a more exact form, and the production process becomes more defined. For example, selecting a requirement that the system be common object request broker architec-

---

[9]   "Project" refers to the product development project.

ture (CORBA)-based adds implementation steps about compiling Interface Description Language (IDL) interfaces, steps for selecting an orb or orb vendor, and changes to the deployment strategy for a product. Cost estimates for the product may also be changed because the expense of licensing an orb must be added to the unit cost of the product.

As variations are selected, the production process is modified to integrate the steps of the attached processes of all of the selected assets into a coherent product-specific production plan. Section 4 presented a discussion on how specific practice areas are determined to be relevant to the production process and how they are sequenced. The individual attached process steps can be integrated into this production plan.

5.2 Developing the Bill of Materials

The bill of materials provides a basis for making cost estimates and schedule predictions for a specific product in a product line. The bill of materials lists all of the assets that are required to build that product. Each asset can be assigned one of the following costs:

- a royalty fee charged on a per-product copy basis
- an amortized internal charge by the developing organization that is a lump sum allocated over the projected number of copies to be sold
- a one-time purchase price from an external source
- no direct charge for the asset

Depending on which initial asset cost is assigned, additional costs may be incurred (e.g., the cost to tailor and test a component for the product). These costs are included in the cost model used to estimate the cost of the product.

The bill of materials also affects schedule prediction. Each asset can be annotated with an availability date, estimated time to modify, or other data that would affect the schedule. This information is combined with the sequencing information defined in the product development process to allow schedules to be created.

The bill of materials can be initialized with entries based on the elements in a sample product implementation, if one exists as a core asset of the product line. Some of the actual components that will be used in the product are known. The entries for sample implementation components are replaced by information about these actual components. Examples include

- externally mandated components. These are components that the product line organization is required to use because of corporate strategies outside the control of the product line developers. These components may be core assets, or they may be specific to a particular product. For example, a subsidiary company may be told by the parent company

that a certain component must be included in the products in a product line as part of an enterprise initiative. In another example, the customer for a specific product may require the use of software created by that customer. That component and the licensing fees associated with it are included in the bill of materials.

- standard, acquired components. These components have such a comprehensive coverage that they are included in every product—as such, these components are core assets. For example, planning for the product line may result in the choice of an external vendor who supplies a portion or portions of every final product. This might be an infrastructure piece used by a number of product companies.

- local core assets. These are the core assets developed by the product line organization. For example, the areas of commonality in the architecture are covered by a standard set of components.

The bill of materials evolves as requirements engineering and architecture definition proceed. This document lists the specific version of each asset being used and provides a link to the product test plan that prescribes interaction tests for the specific combination of assets being used in the product.

5.3 Management Estimates

The production plan includes the initial schedule and preliminary cost estimates based on the resources required to staff the schedule. This information is based on the product line architecture and will be updated after any product-specific architecture definition is completed. The time and costs estimates are constructed using a standard effort rate table that is calibrated to the organization and product line. Techniques such as those used in the Personal Software Process[SM] (PSP[SM]) and Team Software Process[SM] (TSP[SM]) techniques may be applied to determine standard management estimates of size and cost [Humphrey 95]. Table 2 shows an example format for such a table for components. Similar tables would be used for other assets (e.g., eliciting new product-specific requirements, tailoring the architecture, and implementing new test cases).

The estimates are updated as the bill of materials is updated. The exact categories of assets evolve over time. The time estimates for each phase within each component type are based on measurements collected during previous product development efforts.

[SM] Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

Table 2:   Rate Table

Component type	Effort required to integrate (in hours):
New component, one-time use	Analysis: Design: Implementation Test:
New component, new core asset	Analysis: Design: Implementation Test:
New variant on a core asset	Analysis: Design: Implementation Test:
Core asset reuse	Analysis: Design: Implementation Test:

5.4   Maintaining the Production Plan

The production plan must be maintained to ensure that it continues to exhibit the qualities described in Section 7.1 as changes occur. New versions of tools, libraries, and environments often require changes in basic procedures. The production plan for a particular product, at the very least, provides pointers to the current procedures for each activity and may include them directly in the plan. For example, specific build scripts are usually included only by reference because they change often. As new scripts are created, pointers must be adjusted to identify the latest version. Instructions for setting paths are often included directly in the plan because they depend upon the operating system. Changes to how paths are set require a new version of the plan, but new paths do not require a new version.

A number of actions may initiate a change in the production plan and its specialized instantiations, including

- changes to requirements
- revision of business priorities
- creation of a new asset
- release of a new version of an asset
- upgrade of a tool
- revision of a process

These events can result in the addition of steps to, or the reordering of steps in, the basic product development process, thereby triggering schedule reevaluation. The change may simply be a modification of information already in the plan or the addition of information to the plan.

The duration of a project affects the strategy for maintaining the production plan. A project that lasts six weeks can freeze the production plan and ignore upgrades and new releases. A project that lasts six months may need to allow changes due to vendors dropping support for a version or a corporation-wide mandated change in tools or process.

A configuration management tool is used to maintain the production plan. Product-specific production plans are tailored versions of the production plan, but they cannot always be upgraded when the production plan is upgraded. The configuration tool maintains the link between a specific version of a product-specific production plan and the version of the production plan from which it was derived (see Figure 7). The product line production plan goes through multiple versions. As new product-specific production plans are created, a configuration is created for that plan that includes the current version of the production plan.

The configuration of the product-specific production plan contains links to the current set of core assets. As these assets are upgraded, a new configuration is created to accommodate this new version. The product line organization establishes a policy about how often these new configurations are created.
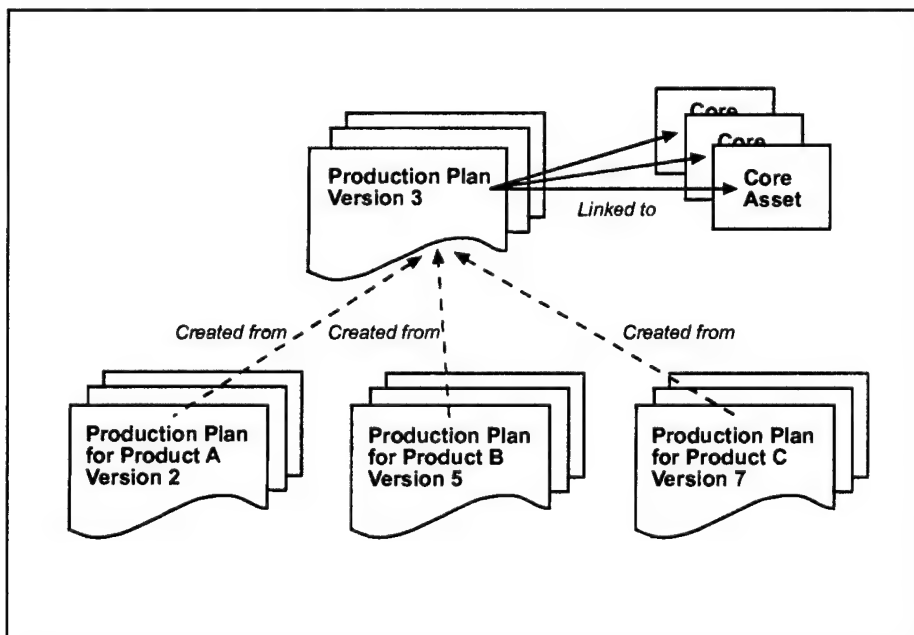
Figure 7: Configurations of Production Plans

6 Using the Production Plan

The production plan is a living document that evolves as it is used. The core-asset developers use the production plan to communicate with the product developers. The product developers use the production plan to guide their day-to-day work. The product developers evaluate the effectiveness of the plan, and the core-asset developers use that evaluation to revise and improve the production plan.

6.1 Interactions

The production plan is used in the context of other concurrent processes. Some of these processes, like the personnel evaluation process, may have little or no interaction with the product development process. Others, such as corporate incentive processes, do interact with the product development process by defining criteria for financial rewards based on performance or the delivery schedule. The production plan provides the necessary interfaces between the product development process and any interacting processes. Two of the most common types of interactions are discussed in the next sections:

- Software Development Processes
- Product Development Processes

6.1.1 Software Development Processes

Organizations often use software development methods such as the Rational Unified Process (RUP) when they initiate a product line approach [Jacobson 99]. These methods have tool support and well-tested approaches to building a piece of software that satisfies a set of requirements. The methods are based on models that define features to address specific development concerns (e.g., iterations and increments to reduce the risk and complexity of development). The production plan serves as the interface between the existing software development method and the product development method of the product line.

The practice areas described in Section 4 supply the activities that populate the software development process. The process described in the production plan modifies the standard process definition to include only those activities needed based on the classification of the product line. The RUP, for example, defines an iterative process in terms of core workflows (i.e., requirements, analysis, design, implementation, and test) and phases (i.e., Inception, Elabora-

---

tion, Construction, and Transition). The core workflows are modified to reflect the characteristics of the product line as illustrated in Section 4.

The production plan defines a different distribution of effort from the usual profile discussed by Jacobson [Jacobson 99]. The Inception and Elaboration phases are reduced as the product line organization gains experience and expertise. The Construction phase is reduced as the market matures and the build process becomes more automated.

In the case of the RUP, the production plan incorporates an iterative, incremental approach. This would be particularly applicable to the example product line in Section 4.1. An iterative and incremental approach helps such an organization cope with an immature market and a lack of product line expertise. As product line practices are institutionalized and the market matures, the need for iterations is reduced. As the build process becomes more automated, the need for increments is also reduced.

6.1.2 Product Development Processes

Software is often part of a larger product such as an embedded system providing part of the functionality of devices such as communication devices, real-time control devices, and home appliances. Product development processes such as the Product and Cycle-Time Excellence (PACE) model define communication interactions among planning, management, and development portions of the organization [McGrath 96]. These approaches extend beyond software development to include hardware development as well as marketing, sales, and maintenance roles.

The production plan defines the interfaces between the process to create the software and the overall product development process. The PACE model, for example, defines a model based on projects that are focused on single products. This definition can be in fundamental conflict with the product line approach. The production plan for building software in an organization using the PACE model defines an interface between the core-asset developers and product developers for each product; this interface is missing from the PACE model. Figure 8 shows that the product developers operate within the normal structure of the PACE model while the core-asset developers work outside the model.
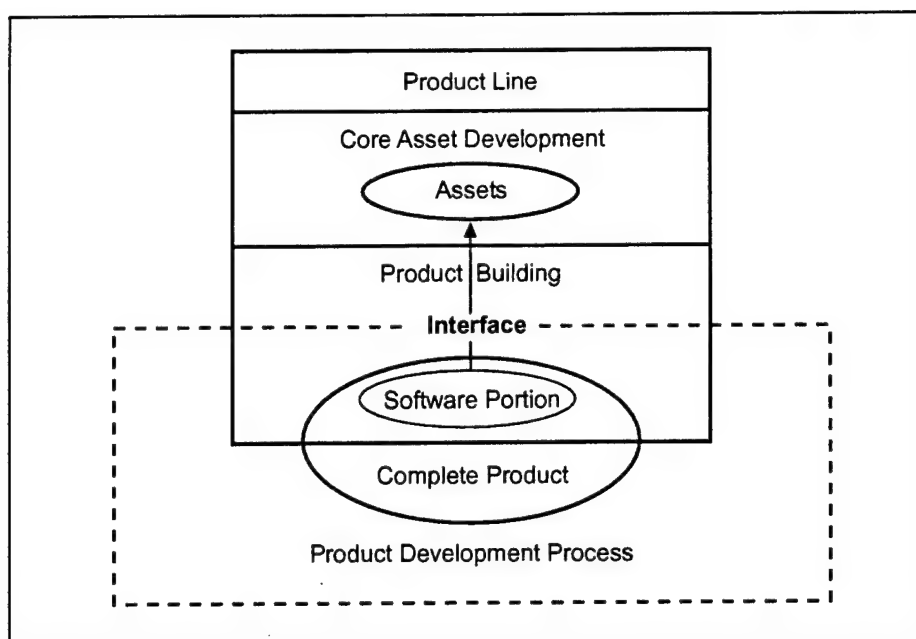
Figure 8: Integration of the Software and Product Development Processes

6.2 Using the Plan Before Product Creation

The product line's production plan provides a basis for planning the construction of a product. The attached process for the production plan is used to create the product-specific production plan. Information in the attached process is used to determine which of the practice areas are needed for developing the product. The plan's attached process steps the product developer through the identification of the core assets associated with each practice area. The attached process for each core asset provides information that supports the development of a schedule for product creation. This information is included in the product-specific production plan.

The information used to instantiate the product-specific production plan can take on many different forms depending on the culture and maturity of the organization. One example is the type of information used in the PSP technique to estimate the size of the final product [Humphrey 95]. The number, size, and complexity of each component are included in a computation that can be used to estimate several product attributes. Table 3 shows the outline for one such calculation.

Table 3:   PSP Categories for Size Estimation

Type of Component				LOC
Base Program	Type	Methods	Relative Size	
Base Size (B)	I/S/L/C	Integer	Very Large – Very Small	
LOC Deleted (D)				
LOC Modified (M)				
Added LOC				
Base Additions (BA)				
New Objects (NO)				
Reused Programs (R)				
Estimated Total LOC	NO + B – D – M + R		=>	

The PSP technique provides detailed planning information while other techniques such as COCOMO offer only system-level estimates [Boehm 81]. The product line will use the technique that best fits its needs.

The product-specific production plan guides the product developers through a process of tailoring the practice areas to the product's needs. Each practice area activity is described in the appropriate attached process. This description includes the core assets needed as well as scheduling information for the resources used to accomplish the activity.

Consider the example in Section 4.2 (highly automated product development, institutionalized product line practices, and a mature market): "Requirements Engineering" and "Testing" are listed as the only practice areas needed for creating products. Since no new development is done for a product in this type of product line, neither the PSP nor COCOMO is necessary. The core-asset team of this product line would work with the product developer teams to define specially designed measures. These measures are then calibrated as the product developer teams gain experience. For example, since most requirements for a product come directly from the product line requirement set, little time is spent in elicitation, analysis, and specification. The most effort is expended clarifying variations. The effort expended on requirements and testing is directly related to the number and complexity of the variation points in the architecture. The product developers calibrate the relationship between variations and effort so that estimates of the time required to determine a product's requirements and to verify their satisfaction improve as the organization gains expertise and experience.

6.3   Using the Plan During Product Development

The product developers execute the production strategy as documented in the product-specific production plan. This plan details the roles and responsibilities of the product developers and provides guidance that can take many different forms.

- The plan guides the product developers through the variability resolution analysis. Each point of variation must be resolved to a specific value. This process typically begins with the selection of specific requirements. These selections are then propagated throughout the product-building process. The resolution may guide the specialization of specific parts of the product line architecture and determine the selection of specific components.

- The plan guides the product developers through the identification of parameter values for generators and constructors. This is provided mainly through the core assets' attached processes. It is also part of the tools description.

- The plan guides managers in conducting reviews, collecting data, and tracking progress. The schedule defined as part of the instantiation of the product-specific production plan contains activities for these management functions.

The plan couples the product-building process and a schedule so that progress can be measured. Structuring the process description and schedule as tables, as shown in the appendix, simplifies the tracking of activities.

6.4 Using the Plan After Product Development

The product developers conduct an "after action" evaluation of the operation of the product line process and the effectiveness of the production plan in guiding that process. The production plan should be structured to facilitate this evaluation. Using tabular descriptions of the process and schedule simplifies the review process. Notes can be captured in a similar table structure during the operation of the process. The evaluation process then steps through the tables in parallel. Problem reports on the assets can be similarly structured.

The product development process is evaluated for characteristics such as clarity, completeness, and correctness. The product developers report to the core-asset developers any changes that are required in the process. Ambiguities in the process are typically resolved during product building, but they are captured again in the After-Action Report.

Each of the assets listed in the bill of materials is also evaluated. The assets are evaluated for their "goodness of fit" with the architecture and with each other. The assets are also evaluated for consistency within the group listed in the bill of materials. A typical inconsistency would be a difference between the implicit specification of an interface attribute, such as timing information in the architecture description, and its realization in code. Any defects in the assets are reported to the core-asset team as quickly as possible, but these are summarized in the evaluation report.

Section 7 provides a detailed view of evaluating the production plan. This evaluation is carried out by the product developers after each use and by the core-asset team. The evaluation

takes place after the production plan is created but prior to its use, and again as After-Action Reports are submitted by product teams.

7 Evaluating the Production Plan

7.1 Characteristics of Good Plans

Jones and Northrop discuss a set of criteria for a good process-improvement action plan [Jones 99]. These criteria fit a production plan equally well. The product line organization evaluates the production plan using these criteria:

- appropriateness for purpose
- clarity
- brevity
- sufficient detail
- internal modularity
- internal and external consistency and traceability
- usability

In the following sections, these criteria are discussed in the context of determining the quality of guidance provided in this report. Each criterion is stated with the assumption that the production plan was written in accordance with the advice given in this report.

7.1.1 Appropriateness for Purpose

The production plan should contain all of the information needed by the product developers. The contents of the plan are appropriate, because they tie the information in the production plan to the core assets that are used to build products. The production plan does not include information about practice areas or assets that are not needed in the product-build process, but it does include information about those practices that are required.

7.1.2 Clarity

The production plan should be understandable to its readers, the product developers. The product developers should be able to build a product from the information given in the plan. The core assets are presented in the production plan to reflect the way in which the product developers use them (see Section 4). This presentation simplifies the instructions needed for

the product developers to be able to locate and effectively use the most appropriate core assets. The example products, described in an appendix of the plan, contribute to this clarity.

7.1.3 Brevity

The production plan should contain only the information that product developers need to construct a product and that cannot be obtained in other places. The process description given in the production plan includes only those practices that the product line allows (see Section 5). For each step in the process, links are given to the attached processes of the appropriate assets rather than repeating the information in the plan. This results in a document that gives a high-level view of the process but provides the information necessary for the product developers to locate the assets they require.

7.1.4 Sufficient Detail

The amount of detail in the production plan is controlled by how complete the attached process of each core asset is. The amount of detail in the plan is adjusted to compensate for information that is missing in the attached processes. Over time, the compensatory information should migrate into the core assets' attached processes. The participation of product developers in constructing and evaluating the plan (see Section 3), ensures that there is sufficient information to support product building.

7.1.5 Internal Modularity

The production plan should be modular to the extent that attached processes are included by reference rather than by content. For example, it should be possible to modify the architecture process without modifying the production plan. The use of practice areas rather than precise process steps (see Section 5) maintains a clean separation between types of activities and makes the plan more modular.

7.1.6 Internal and External Consistency and Traceability

The production plan should be consistent in two ways. First, it should maintain external consistency by keeping only pointers to the attached processes. Second, it should maintain internal consistency by selecting the process steps from the overall process illustrated in Figure 9 in the appendix. Actions taken by the product developers can be traced directly to the processes attached to specific core assets.

7.1.7 Usability

The production plan is most usable if it is written from the product developers' perspective. The process descriptions in Section 4 and the appendix provide the product developers with detailed sequences of activities that structure the product-building process.

7.2 Evaluation Metrics

Associated with each characteristic is information to evaluate the degree to which the plan possesses the quality. This information is collected from the product developers and from observation of the product-creation process.

- appropriateness of purpose: Does the plan contain steps that can be eliminated without affecting the product being developed?
- clarity: Which parts of the plan are the sources of the most requests for information from the core-asset team?
- sufficient detail: Was any information used that was not referenced in the production plan?
- brevity: What information is contained in the production plan but never used?
- internal modularity: When a change is made to the product plan, how many parts of the plan are affected?
- internal and external consistency and traceability: When a core asset is modified, is it possible to identify where changes should be made in the production plan? Are any of the problem reports about the production plan due to inconsistencies between the information in the plan and in the core asset?
- usability: How often is the information requested from the core-asset team already in the product plan?

7.3 Evaluating the Plan

The production plan is evaluated periodically. The start of a new product development effort is a particularly useful time at which to review and evaluate the plan. Specifically, reviewing the accuracy of the schedules and cost estimates from previous projects should be done at the start of a new product, before the same calibrations are used to produce new estimates. The product developers also review the plan after a product is delivered, as discussed in Section 6.4.

The plan should also be evaluated whenever there is a major revision in any of the key core assets. For example, if there is a release schedule for versions of the architecture, there should be a corresponding schedule to review the validity of the production plan. The release of ma-

jor revisions of component libraries or tool upgrades should also trigger evaluation of the validity of the product-building process.

The core-asset developers and the product developers jointly evaluate the production plan. The product developers evaluate the plan in terms of the characteristics described in Section 7.1. The results of that evaluation are provided to the core-asset developers, who evaluate the information in the plan to ensure that it reflects the current version of each core asset. The core-asset developers then update the production plan and any other pieces that need modification.

8 Future Work

More experience is needed in creating and utilizing production plans for software product lines. While hard goods manufacturers have a long history of creating production plans, this is a relatively new concept for software developers. Much of the information in a production plan is produced in some form by organizations that don't have product lines. By studying the content, scope, and depth of production plans in hard goods manufacturing, the production information captured by organizations both with and without product lines can be improved.

There is information available in a product line organization that is typically not available in those without product lines. One example is the production strategy. The core-asset developers must communicate their prescription for how products are to be produced. Work is needed to determine the best approach for this communication. Is it a process description, or is it more narrative?

The core-asset developers create the production plan for the product developers. In some organizations, the two groups may have very different levels of domain and development expertise. In other organizations, there is no clear separation between the core-asset developers and product developers. Work is needed to determine exactly how the core-asset developers can understand the product developers' perspective and produce a document that is written from that perspective.

A number of questions still need to be answered:

- How should software production plans be structured to achieve the maximum benefit?
- What effect do the variations among the products have on the production plan?
- Can existing project management tools be used to instantiate, operate, and track the product-building process?
- We identified the need for the bill of materials, but the description was incomplete. What information would be useful besides cost and schedule?

More empirical evidence is needed to further clarify the role of the production plan and to provide more concrete guidance. The Software Engineering Institute (SEI) will continue to investigate these issues as it conducts research, collaborates in product line efforts, writes case studies about product lines, and participates in workshops and product line forums.

Appendix   Practice Areas of the Product Builder Pattern

The Product Builder Pattern describes the interaction of the practice areas needed to build a product [Clements 02a]. Figure 9 shows the practice areas required by the Product Builder Pattern and shows interactions among them in terms of information flow. In Section 4, three examples are given in which a subset of the practices is used depending upon the location of the product line in the classification scheme shown in Figure 2. In this appendix, each of the practices in the full set is explained.
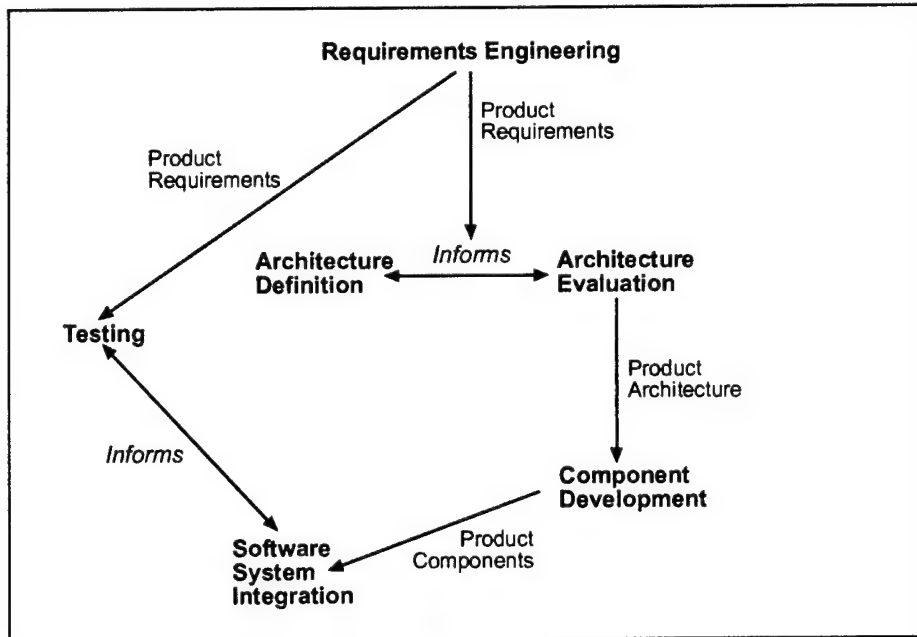
Figure 9:   The Product Builder Pattern

The following subsections contain a summary for each practice area as if it were a single step in a process. The summary is illustrative rather then prescriptive. For example, responsibilities differ from organization to organization, and not all organizations use the specific techniques mentioned. The summary follows the style used by Russ [Russ 00]. This level of information is sufficient for the purpose of describing the production plan. Clements provides additional information about each area [Clements 02a].

Requirements Engineering

Every production plan includes "Requirements Engineering" as a required practice area, because every product has specific—and to some degree unique—requirements. Most of the product's requirements are derived from the product line requirements, with choices being made at each variation point. Where there are deviations from the product line requirements, new requirements are written. Table 4 summarizes the external interfaces of the "Requirements Engineering" practice area.

An individual requirement is the fundamental unit of traceability. Test cases, incremental development plans, and phased product deliveries are all associated with specific requirements. These pieces are identified as the requirements set for a product being created.

The product developers' requirements engineering process includes selecting a set of requirements that is consistent and complete. A tool that uses predefined relationships between groups of requirements can manage the selection process. In response to selections made by a product developer, the tool guides the inclusion of all requirements within a group.

Alternatively, this process may be a manual process. Diagrams that show dependencies between requirements facilitate the manual selection of a product's requirements from the larger product line requirements set. For example, a Unified Modeling Language (UML) use-case diagram shows a variety of dependencies between the use cases. Figure 10 illustrates that if use-case C is to be included in a product's requirement set, other related use cases must be selected as well.
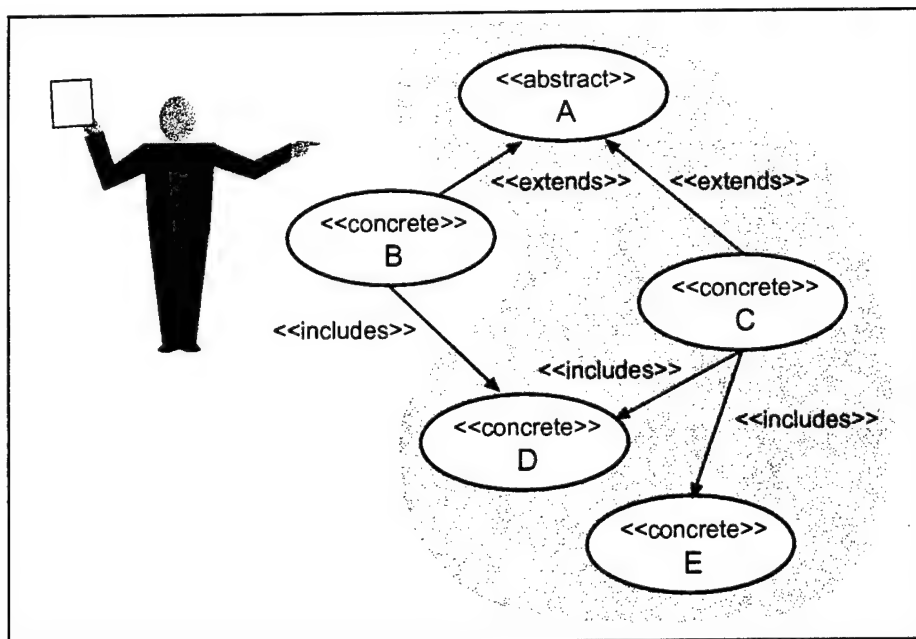
Figure 10: Use Case Dependencies

Table 4:   Phase 1—Requirements Engineering

Section Heading	Section Content
Description	The product developers develop the complete list of requirements for the specific product. The list is selected from the product line's pre-defined set. Additional requirements may be added if approved by the appropriate authority.
Responsibility	The product team has responsibility for being certain that a consistent set of requirements has been chosen from the product line's requirements. The core-asset developers are responsible for ensuring that a sufficiently comprehensive set of requirements is available from which to select.
Input	The customer selects a product and designates specific values for optional features.
Entry Criteria	The development of a new product has been approved as within the scope of the product line organization.
Activities	Requirements are selected from the product line set. Additional requirements are added. The requirement set is evaluated for completeness, correctness, and consistency.
Output	A comprehensive description of the requirements for the product is created.
Exit Criteria	If the requirements set is "complete enough" for examining the architecture, the next stage begins.

The validity of the product line analyses may be questionable if a sufficient amount of the requirements fall outside the product line set. In this case, the product approval process should be revisited. |
| Metrics | The percentage of product requirements already belonging to the product line's requirement set is calculated. |

Architecture Definition

Products that introduce new requirements beyond the product line require additional architecture definition work. If the deviation from the product line's set of requirements is minor, the architect may be able to modify the architecture definition directly. If the deviation is major, the architect may wish to conduct an architecture evaluation prior to making final modifications to the architecture. Table 5 summarizes the external interfaces of the "Architecture Definition" practice area.

The selection of specific requirements for a product results in certain portions of the architecture being selected as well. In particular, certain variants are selected at architecture variation points. If new requirements have been added, new variants may be defined at some of the variation points.

Table 5: Phase 2—Architecture Definition

Section Heading	Section Content
Description	New architecture structures may be created, or existing structures may be modified. In this case, the technique is applied incrementally to add the functionality for this specific product.
Responsibility	The product architecture team is responsible for making changes and additions.
Input	The input is the requirement set, including any requirements not satisfied by the current architecture.
Entry Criteria	The architects can begin modifying the architecture whenever additional requirements have been accepted into the project.
Activities	Architectural styles are applied, and component interfaces are defined.
Output	The architecture is revised and expanded to satisfy additional requirements.
Exit Criteria	It is possible to exit this phase when all of the new requirements, including quality attributes, have been satisfied.
Metrics	The standard, quality attribute measures are computed.

The assets and tools available to the product developers may not allow new requirements. For example, the product developer may select from an automated checklist of requirements, in which case, the architecture definition activity is not part of the process.

Architecture Evaluation

Any new architecture definition requires corresponding evaluation. As mentioned in "Architecture Definition" (above), if new requirements have been proposed that result in a major modification of the architecture, the architects may decide to use the Architecture Tradeoff

Analysis Method[SM] (ATAM[SM]) to investigate any proposed definitions [Clements 02b]. Table 6 summarizes the external interfaces of the "Architecture Evaluation" practice area.

An incremental evaluation is performed that covers those portions of the architecture where a new definition or a modification has occurred. The ATAM scenarios can be associated with specific requirements. A set of scenarios is identified for use when a requirement is selected for inclusion in a product.

The Architecture Evaluation phase will not be included in the product-building process if neither new requirements nor new architecture definitions are allowed. In this case, products are defined by choosing from a fixed set of requirements.

Table 6: Phase 3—Architecture Evaluation

Section Heading	Section Content
Description	The ATAM is used to evaluate the degree to which the architecture satisfies its requirements. In this case, the evaluation is performed incrementally on anything that is being changed from the product line architecture.
Responsibility	The architecture team initiates the evaluation process that involves both core-asset developers and product developers.
Input	An architecture description and a set of requirements that the architecture must satisfy make up the input.
Entry Criteria	This step is entered when a change is either proposed or actually made to the product line architecture.
Activities	Scenarios that are valid uses of the software are created and used to investigate both new and preexisting portions of the architecture.
Output	Defects in the architecture are found and listed.
Exit Criteria	This phase can be exited when sufficient scenarios have been investigated to cover new requirements.
Metrics	The percentage of the architecture that has been touched during the simulated execution of the software is computed.

Component Development

Component development includes the tailoring of existing components and the development of new components. New component development may be required even if no new architecture definition has occurred. A particular variant may have been foreseen in the architecture, but no implementation may have been created for that specific component. When a need arises, the component is built. Initial or tailored versions of a component will be tested rigorously but may require several iterations before their attributes satisfy the specified values. Table 7 summarizes the external interfaces of the "Component Development" practice area.

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

As the product line and the market mature, the need for basic component development declines. The potential for acquiring components increases as more vendors enter the market. The product line's inventory of acceptable components also increases as more variants are implemented.

Component development is not a part of the product-build process if the product line is constrained to a set of existing configurations. When component development is in the process, the plan specifies how development tools are used to ensure compatibility with the existing component inventory.

Table 7: Phase 4—Component Development

Section Heading	Section Content
Description	Implementations of the required specifications are created. A component may be assigned specific quality attribute targets including a portion of the performance budget.
Responsibility	The implementation specialists among the product developers develop or tailor the components. The component may be acquired rather than built.
Input	The specifications from the architecture description plus a list of needed components and changes to existing components comprise the input.
Entry Criteria	The architecture description is sufficiently complete to support initial efforts at component development and alteration.
Activities	Available products are found and purchased, if feasible. Components are designed and implemented either from scratch or by modifying existing components.
Output	Initially, candidate implementations and alterations of specified components result. Eventually, mature implementations that achieve expected quality values result.
Exit Criteria	Components are accepted for incorporation into the product.
Metrics	Component reliability and testability measures, performance measures, and other quality attribute requirements are computed.

Testing

Every production plan prescribes a set of testing activities that is tailored to the specific needs of the product line. The "Testing" practice area defines several levels and types of testing. Table 8 summarizes the external interfaces of that practice area. The first level of testing that is applied is basic unit testing of each completed component. The process diagram in Figure 9 shows that testing may produce feedback to component development. This iteration continues until the component meets its functional and quality requirements. When no new component development is allowed during product building, no unit test activity is specified.

Components are the inputs to the "Software System Integration" practice area where they are incorporated with other components. Various types of interaction tests are then performed to

ensure that the integrated pieces continue to meet requirements. Eventually, all the components needed to satisfy all of the system's software requirements are integrated and tested.

Table 8:   Phase 5—Testing

Section Heading	Section Content
Description	Variously sized pieces of software are executed with test cases for which correct answers are known. The behavior during execution is observed, and a judgment is made about the quality of the software.
Responsibility	Various persons are responsible depending upon the type of testing being discussed.
Input	The input to each test activity is the item to be tested and the specification for that item.
Entry Criteria	The item has reached sufficient stability to test some portion of the item.
Activities	Specification to select test cases is analyzed. Test cases are constructed and executed. Test results are analyzed.
Output	The test cases applied during testing as well as the test results that indicate whether the item passed or failed each test case are provided.
Exit Criteria	If sufficient test cases pass, the next phase begins. If many test cases fail, the previous phase is revisited.
Metrics	The percentage of test cases passed and the percentage of the item covered by tests are computed.

Software System Integration

Software system integration combines components to form a system or subsystem. This is often the main focus of product building. The production plan specifies how this integration can occur. Table 9 summarizes the external interfaces of the "Software System Integration" practice area.

Table 9:   Phase 6—Software System Integration

Section Heading	Section Content
Description	The product is constructed by integrating components.
Responsibility	The product developers are responsible for integrating the components.
Input	The tested components and the product architecture are inputs.
Entry Criteria	Some components have passed testing.
Activities	The glue code is written, and the integrated subsystem is built.
Output	An integrated subsystem is the output.
Exit Criteria	The subsystem has passed interaction tests.
Metrics	The percentage of requirements satisfied is calculated.

The integration can result in a new arrangement of components. As these new configurations are constructed, they are fed back to the "Testing" practice area to be verified.

Section 6 discusses how the techniques and activities needed to integrate the components are described in the production plan. This may include altering configuration files, property files, and build scripts.

Product lines that automatically generate the product from a selected list of requirements do not utilize the "Software System Integration" practice area. In this case, the automatically generated products are tested at a system level and may incorporate other post-development testing techniques, such as user acceptance testing.

References

[Bass 98]   Bass, Len; Clements, Paul; & Kazman, Rick. *Software Architecture in Practice*. Boston, MA: Addison-Wesley, 1998.

[Batory 97]   Batory, Don. *Intelligent Components and Software Generators* (Technical Report 97-06). Austin, TX: Department of Computer Sciences, University of Texas at Austin, 1997.

[Boehm 81]   Boehm, Barry. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[Clements 02a]   Clements, Paul & Northrop, Linda. *Software Product Lines: Practices and Patterns*. Boston, MA: Addison-Wesley, 2002.

[Clements 02b]   Clements, Paul; Kazman, Rick; & Klein, Mark. *Evaluating Software Architectures: Methods and Case Studies*. Boston, MA: Addison-Wesley, 2002.

[Cohen 99]   Cohen, Sholom. *Guidelines for Developing a Product Line Concept of Operations* (CMU/SEI-99-TR-008, ADA367714). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu/publications/documents/99.reports/99tr008/99tr008abstract.html>.

[Hax 87]   Hax, Arnoldo. "Aggregate Production Planning." *Production Handbook, Fourth Edition*. New York, NY: John Wiley & Sons, 1987.

[Humphrey 95]   Humphrey, Watts. *A Discipline for Software Engineering: The Complete PSP Book*. Boston, MA: Addison-Wesley, 1995.

[Jacobson 99]   Jacobson, Ivar; Booch, Grady; & Rumbaugh, James. *The Unified Software Development Process*. Boston, MA: Addison-Wesley, 1999.

[Jones 99]   Jones, Larry & Northrop, Linda. "Software Process Improvement Planning." *Fourth Annual European Software Engineering Process Group Conference.* Amsterdam, The Netherlands, June 7 - 10, 1999. Milton Keynes, UK: European Software Process Improvement Foundation, 1999.

[Kang 90]   Kang, Kyo C.; Cohen, Sholom G.; Hess, James A.; Novak, William E.; & Peterson, A. Spencer. *Feature-Oriented Domain Analysis Feasibility Study* (CMU/SEI-90-TR-21, ADA235785). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990.

[McGrath 96]   McGrath, Michael E. *Setting the PACE in Product Development.* Woburn, MA: Butterworth-Heinemann, 1996.

[Russ 00]   Russ, Melissa L. & McGregor, John D. "A Software Development Process for Small Projects." *IEEE Software 17*, 5 (September-October 2000): 96-101.

[Weiss 99]   Weiss, David M. & Lai, Chi Tau Robert. *Software Product-Line Engineering.* Reading, MA: Addison-Wesley, 1999.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE June 2002	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Guidelines for Developing a Product Line Production Plan		5. FUNDING NUMBERS F19628-00-C-0003
6. AUTHOR(S) Gary Chastek, John D. McGregor		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2002-TR-006
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2002-006

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS	12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

A production plan is a description of how core assets are to be used to develop a product in a product line. A product line organization creates such a plan to ensure that the correct core assets are used appropriately to build a specific product in a specific way. The production plans and techniques used to create products vary widely from organization to organization and from one product line to another. Because of this variance, the developers of production plans need some guidance about the plans' form and content.

This technical report provides guidance for creating, using, and evaluating a production plan. In addition, this report presents a classification scheme that describes the characteristics of a product line organization that influence the form and content of the production plan.

14. SUBJECT TERMS Product Line, Production Plan, Process, Product Developer	15. NUMBER OF PAGES 63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102